

Kotlin

Tien Pham



Kotlin?

Kotlin

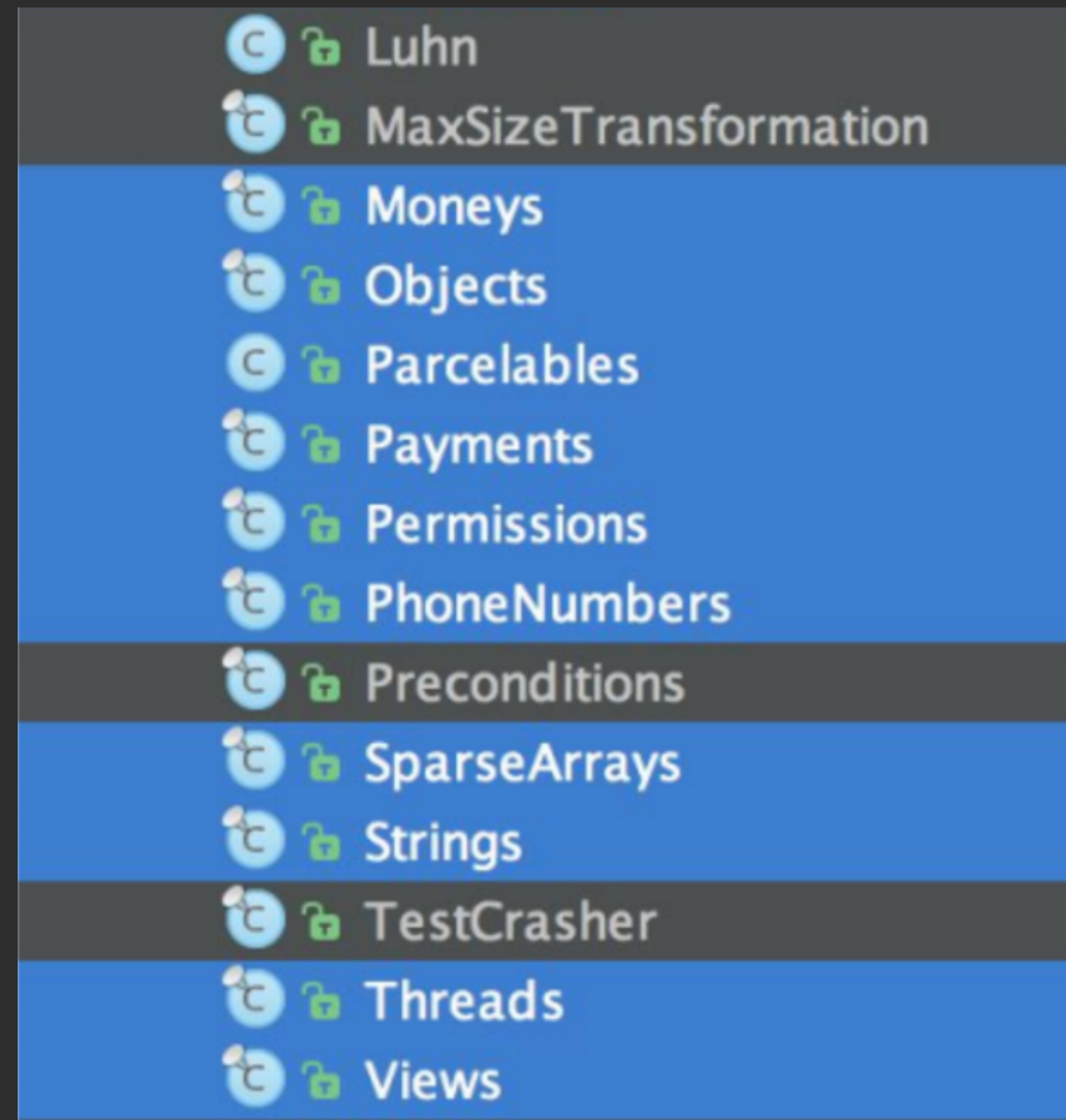
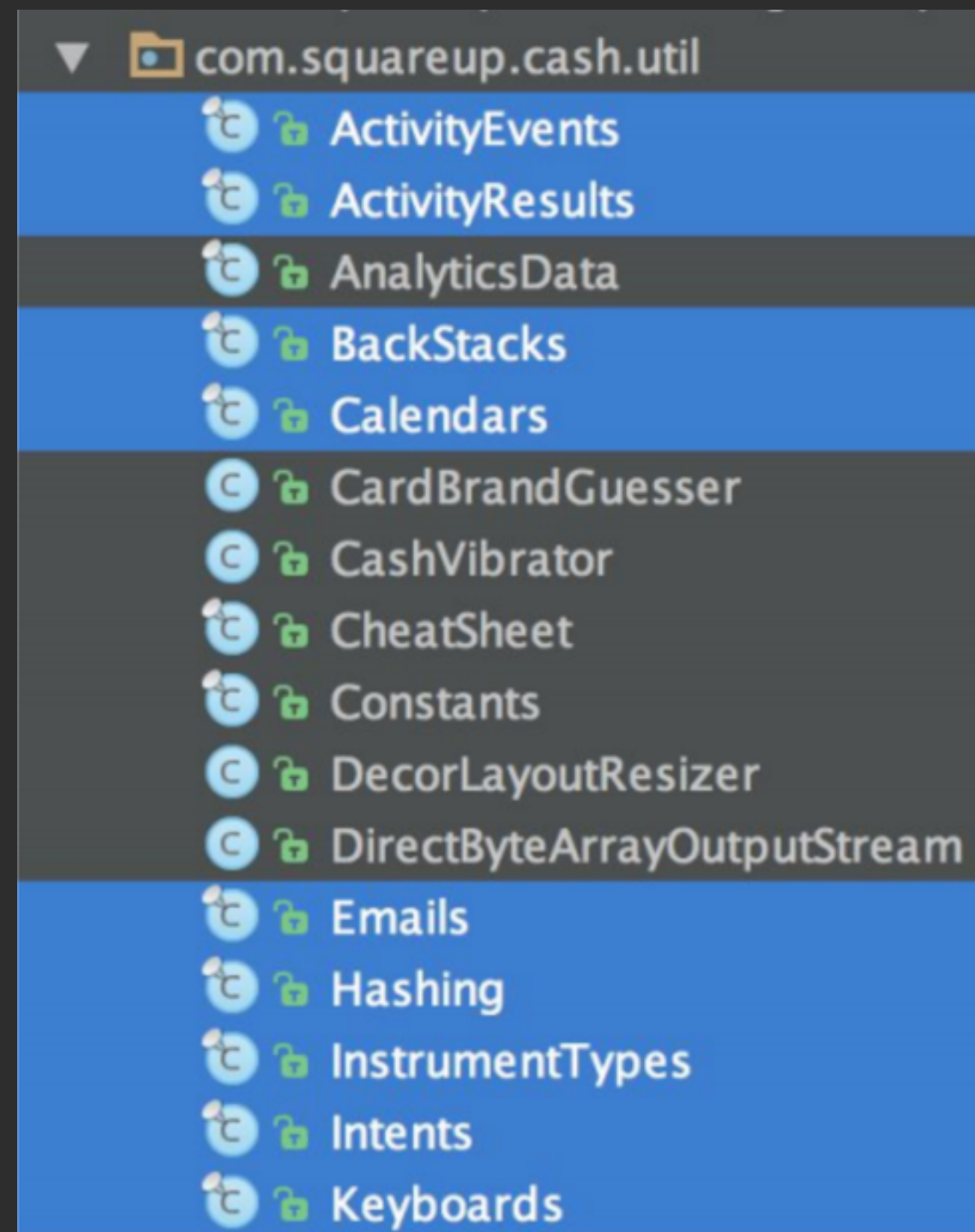
- JVM based language
- JetBrains, IntelliJ IDEA, heard of them?
- Focus on readability, correctness, and developer productivity

Kotlin

- Lightweight (<7000 methods)
- Easy to learn
- Highly interoperable
- Perfectly integrated with Android Studio and Gradle

Extension Methods

Extension Methods



Extension Methods

- Kotlin allows the declaration of both static and instance methods into types which you do not control.
- Kotlin doesn't eliminate utility methods - it gives them superpowers!

Extension Methods

```
fun String.last() : Char {  
    return this[length() - 1]  
}
```

```
val x = "Hey!";  
println(x.last())
```


Extension Methods

```
fun String?.isNullOrEmpty() : Boolean {  
    return this == null || this.length() == 0  
}
```

```
val nullString: String? = null  
println(nullString.isNullOrEmpty())
```

```
fun View.shake() {  
    ObjectAnimator.  
        ofFloat(this, "rotation",  
            0f, 10f, -10f, 6f,  
            -6f, 3f, -3f, 0f)  
        .setDuration(400).start()  
}
```

```
myTextView.shake()  
myButton.shake()
```

```
...
```

Lambdas

Lambdas

```
fun saySomething(name: String, f: (String) -> Unit) {  
    f(name)  
}
```

```
saySomething("Kotlin",  
{name -> println("Hello $name")})
```

Lambdas

```
someButton.setOnClickListener({ finish() })  
someButton.setOnClickListener { finish() }
```

```
users.filter {it.age > 21}  
      .sortedBy {it.lastName}  
      .map {it.id}
```

Null Safety

Null Safety

- Kotlin is null-safe!!!

Null Safety

```
var artist: Artist? = null
var notNullArtist: Artist = null
```

```
artist?.print()
artist.print()
```

```
if (artist != null) {
    artist.print()
}
```

```
// Use Elvis operator to give an alternative in case the object is null
```

```
val name = artist?.name ?: "empty"
```

```
// Only use it when we are sure it's not null.
```

```
// Will throw an exception otherwise.
```

```
artist!!.print()
```


Data classes

- Classes that do nothing but hold data

```
data class User(val name: String, val age: Int)
```

Implicit casting

```
var x: Object = "Hi"  
val length = x.length() // Doesn't compile  
  
if (x is String) {  
    x.length() // Compiles!!!  
}
```

Implicit casting

```
var x: Object? = "Hi"  
  
if (x != null) {  
    val y: Object = x // x is a non-null type here  
}
```

Lazy properties

```
var p: String by Delegate()  
val lazyValue: String by lazy { expensiveOperation() }
```

But Scala is way better!



“Tiếng Tây Tiếng Ta” là một
game đoán chữ giúp bạn dễ nhớ
từ vựng tiếng Anh hơn hoặc là
không

Q&A